

**EXAMINER'S AMENDMENT**

1. . An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. Authorization for this examiner's amendment was given in a telephone interview with Mr. William J Kramer (Reg. No. 46229) on 19 March 2008.

3. The claims had been amended as *follows*:

1. (Currently Amended) A method for tracking kernel resource usage comprising :

generating a tag to charge a process allocated with kernel resources wherein generating a tag further comprises:

determining whether a request for kernel resources is passed from an intermediate function using a worker thread;

finding at least one link between the worker thread and the process; and,

identifying the process that originated the request according to the found link between the worker thread and the process;

determining whether the process is a kernel process;

determining whether the process is a user process;

flagging the tag with a kernel flag to indicate whether the process has been determined to be a kernel process, the flagging further comprising:

generating a tag value to identify kernel resources allocated to a user process;

saving a type of kernel resources allocated to the tag; and,

saving a user process identifier to the tag to identify the process;

flagging the tag with a user flag to indicate whether the process has been determined to be a user process based upon the determination step wherein the tag value and the type of kernel resources are saved in a first word of the tag, and the user process identifier is saved in a second word of the tag;

determining whether the process is a first predefined process or a second predefined process;

saving an identifier to the tag to identify whether the process is a first predefined process or a second predefined process based upon the determination step;

determining whether the process has used more kernel usage than a threshold; and

if the process kernel usage is above a threshold, aborting the process.

2. (Canceled)

3. (Currently Amended) The method of claim 1 wherein flagging the tag further comprises the steps of:

generating a tag value to identify the kernel resources allocated to a kernel process; and,

saving a driver identification to the tag value.

4. (Original) The method of claim 3 wherein the tag value with the driver identification is saved in a first word of the tag.

5. (Canceled)

6. (Canceled)

7. (Currently Amended) The method of claim 1 wherein ~~said step of~~ flagging the tag further comprises:

generating a tag value to identify kernel resources allocated to a user process;

saving the tag value to a first word of the tag; and,

saving a user process identifier to identify the process to a second word of the tag.

8. (Canceled)

9. (Canceled)

10. (Canceled)

11. (Canceled)

12. (Canceled)

13. (Canceled)

14. (Canceled)

15. (Canceled)

16. (Canceled)

17. (Canceled)

18. (Canceled)

19. (Canceled)

20. (Canceled)

21. (Canceled)

22. (Canceled)

23. (Canceled)

24. (Canceled)

25. (Canceled)

26. (Canceled)

27. (Canceled)

28. (Canceled)

29. (Canceled)

30. (Currently Amended) A computer storage medium having computer-executable instructions to be executed by a processor\_for performing a method for tracking kernel resource usage comprising:

generating a tag to charge a process allocated with kernel resources wherein generating a tag further comprises:

determining whether a request for kernel resources is passed from an intermediate function using a worker thread;

finding at least one link between the worker thread and the process; and,

identifying the process that originated the request according to the found link between the worker thread and the process;

determining whether the process is a kernel process;

determining whether the process is a user process;

flagging the tag with a kernel flag to indicate whether the process has been determined to be a kernel process, the flagging further comprising:

generating a tag value to identify kernel resources allocated to a user process;

saving a type of kernel resources allocated to the tag; and,  
saving a user process identifier to the tag to identify the process;  
flagging the tag with a user flag to indicate whether the process has been  
determined to be a user process based upon the determination step wherein the tag  
value and the type of kernel resources are saved in a first word of the tag, and the user  
process identifier is saved in a second word of the tag;  
determining whether the process is a first predefined process or a second  
predefined process;  
saving an identifier to the tag to identify whether the process is a first predefined  
process or a second predefined process based upon the determination step;  
determining whether the process has used more kernel usage than a threshold;  
and  
if the process kernel usage is above a threshold, aborting the process.

31. (Canceled)

32. (Canceled)

33. (Canceled)

34. (Canceled)

35. (new) The computer storage medium of claim 30, wherein flagging the tag further comprises the steps of:

generating a tag value to identify the kernel resources allocated to a kernel process; and saving a driver identification to the tag value.

36. (new) The computer storage medium of claim 35, wherein the tag value with the driver identification is saved in a first word of the tag.

37. (new) The computer storage medium of claim 36, wherein flagging the tag further comprises of:

generating a tag value to identify kernel resources allocated to a user process;  
saving the tag value to a first word of the tag; and  
saving a user process identifier to identify the process to a second word of the tag.

38. (new) The computer storage medium of claim 30, further comprising:  
saving a process identifier to identify the process allocated with the kernel resources; and  
saving a type of kernel resources allocated to the process.

***Allowable Subject Matter***

4. Claims **1, 3-4, 7, 30 and 35-38** are allowed.

a. As to **claim 1**, the prior art of record does not expressly teach determining whether a request for kernel resources is passed from an intermediate function using a worker thread; finding at least one link between the worker thread and the process; and, identifying the process that originated the request according to the found link between the worker thread and the process; flagging the tag with a kernel flag to indicate whether the process has been determined to be a kernel process, the flagging further comprising: generating a tag value to identify kernel resources allocated to a user process; saving a type of kernel resources allocated to the tag; and, saving a user process identifier to the tag to identify the process; flagging the tag with a user flag to indicate whether the process has been determined to be a user process based upon the determination step wherein the tag value and the type of kernel resources are saved in a first word of the tag, and the user process identifier is saved in a second word of the tag when taken in the context of the claim as a whole. More over, the art of record does not provide a basis of evidence for asserting a motivation driven from the art or from one knowledgeable in the art, that one of ordinary skill in the art at the time the invention was made would have modified a method for tracking kernel resource usage to combine the disclosed limitations as recited in the context of **Claim 1**.

b. As to **Claim 30**, being directed to a product for tracking kernel resource usage having substantially the same limitations as **Claim 1**, this claim is allowable for the same reasoning as recited in **Claims 1** above.



Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Exr. Abdou Seye whose telephone number is (571) 270-1062. The examiner can normally be reached Monday through Friday from 7:30 a.m. to 4:00 p.m.

If attempts to reach the examiner by telephone are unsuccessful, contact the examiner's supervisor, An Meng at (571) 272-3756. The fax phone number for formal or official faxes to Technology Center 3600 is (571) 273-8300. Draft or informal faxes, which will not be entered in the application, may be submitted directly to the examiner at (571) 273-6722.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group Receptionist whose telephone number is (571) 272-3600.

AKS

March 19, 2008

/Meng-Ai An/

Supervisory Patent Examiner, Art Unit 2195